2013 International Conference on Indoor Positioning and Indoor Navigation, 28-31st October 2013

On-board navigation system for smartphones

Mauricio César Togneri
Institute of Services Science
University of Geneva
Switzerland
mauricio.togneri@unige.ch

Michel Deriaz
Institute of Services Science
University of Geneva
Switzerland
michel.deriaz@unige.ch

Abstract—**Several mobile solutions offer the possibility to download maps to use them offline at any moment. However, most of the time a connection to an external server is still needed in order to calculate routes and navigate. This represents an issue when traveling abroad due to roaming costs. In this paper, we propose a solution to this problem through an engine that stores and manages OpenStreetMap's data to consult points of interest, calculate routes and navigate without any connection required. The software manages indoor and outdoor information to provide a full navigation service that works in both environments. Therefore, the same system allows navigating in a highway by car and provides indoor navigation for museums, hospitals and airports among others. The result is an on-board engine for smartphones that provides indoor and outdoor navigation services that does not require Internet connection.**

Keywords: on-board, navigation, indoor, outdoor, smartphones

I. INTRODUCTION

Nowadays we can find several web mapping services of which we can highlight Google Maps[1], Bing Maps[2] or Nokia Here[3] among others. All these solutions also provide online navigation services and represent a very important tool in several situations. However, we may encounter situations with specific constraints where an Internet connection is not allowed or guaranteed. In this case we cannot rely on online services and we are forced to use a solution where the navigation services work offline.

Another important aspect of the web mapping services is their large coverage range, mostly at worldwide level. This characteristic allows us to take a look at almost every corner in the world and calculate routes between two points that are thousands of kilometers away from each other. Nonetheless, most of the biggest providers have a closed source of information that we cannot change or access freely. There are some exceptions like MapShare[4] from TomTom or MapMaker[5] from Google that allow users to modify certain parts of the map. However, users do not have the rights over the edited maps and all contributions become property of the companies (map information will remain proprietary and not free).

[1] Google Maps, http://www.google.com/maps
[2] Bing Maps, http://www.bing.com/maps
[3] Nokia Here, http://www.here.com
[4] TomTom MapShare, www.tomtom.com/mapshare
[5] Google MapMaker, http://www.google.com/mapmaker

One exception to this problem is OpenStreetMap[6], a collaborative project to create a free editable map of the world that provides geographical data to anyone. Thanks to this project, users can access freely to a world map, modify it and create their own maps.

Another aspect of the navigation systems is the indoor maps availability. This topic is relatively new and most of the solutions do not provide a large coverage for indoor navigation. Although it is possible to find indoor services in important places (e.g., airports, big shopping malls, etc.), users still rely on providers to have access to indoor navigation.

Therefore, our goal is to create a navigation system that uses a source of information that is free, continuously growing and easy to modify, a system that works offline and allows navigating both indoor and outdoor. In this paper we present a solution that takes advantage of OpenStreetMap's data and format to create a navigation system for smartphones. This approach solves the problem of service unavailability that users can have due to a lack of Internet connection and allows indoor and outdoor navigation.

The result is a generic navigation system that can be integrated in different situations, for example:

- For touristic purposes, in a guide application to explore a new city (offline feature).
- As a customized navigation system for museums, hospitals, airports, etc. (indoor feature).
- Customized car navigation for companies to control their fleet (personalization feature).
- As a route planner for emergency situations (high availability feature).

The remaining sections are structured as follows. Section II provides an overview of the related work in the area of mobile navigation systems. Section III describes the main architecture and implementation of the module. In Section IV we present an example application that shows the services provided by the system. Finally we present our conclusions and future work in Section V.

[6] OpenStreetMap, http://www.openstreetmap.org

II. Related Work

From a market perspective, most of the mapping solutions offer the possibility to download maps to use them offline at any moment. The best known example is Google Maps and its mobile application[7] that allows users to download maps on the phone. However, this can help us to locate ourselves over the map but it is not able to provide offline navigation services because a request to the server needs to be done.

There are some mobile solutions such as Sygic: GPS Navigation & Maps[8] or ROUTE 66 Maps + Navigation[9], which allow downloading maps and calculate routes offline. Even so, those kinds of applications are closed, not free and users have no control over the map that they are using. They are allowed to report bugs or problems on the roads but it is not guaranteed that the changes will be applied.

Some other mobile applications such as Navfree: Free GPS Navigation[10] and OsmAnd Maps & Navigation[11] solve the previous problem. Both are a good example of a free application that uses OpenStreetMap as a source of maps. Even though, these applications do not offer indoor navigation.

From a research perspective, Jiang, Fang, Yao and Wang [1] present a full infrastructure to deploy an indoor & outdoor navigation system. However, the model relies on a network architecture that uses servers to provide the navigation services. This solution also uses a specific handheld device which makes it difficult to implement the system in a real situation.

Moreover, in the work of Li and Gong [2] we find another attempt to create a system that integrates indoor and outdoor navigation. Nevertheless, we encountered the same problem of a server connection dependency. In this case the system uses the Google Maps API to acquire the routes for outdoor and it uses a local server to provide the product's querying services and the indoor route calculation.

The novelty of our work is a software module called NaviMod (Navigation Module) that solves all the previous problems. In other words, it is a system that works offline, uses an open, free and customizable source of maps and allows both indoor and outdoor navigation.

III. Architecture and Implementation

The entire system consists of an Android library, which means that it can be integrated in a variety of devices, such as smartphones, tablets, smart watches and the future Google Glass among others.

A. Requirements

The system assumes that the final application has access to a position provider due to the navigation system requiring the user's location in order to perform certain navigation services.

[7] Google Maps Mobile, http://www.google.com/mobile/maps
[8] Sygic: GPS Navigation & Maps, http://www.sygic.com
[9] ROUTE 66 Maps + Navigation, http://www.66.com
[10] Navfree, http://www.navmii.com/gpsnavigation
[11] OsmAnd Maps & Navigation, http://www.osmand.net

The final application acquires the user's location from the position provider and sends it to the navigation module as it is shown in figure 1. This external module provides the device's position using the World Geodetic System, revision WGS 84. Inside of the navigation module, all points on the map as well as the user's position are represented using latitude, longitude and altitude following the specified coordinate system. Therefore, every provider should be compatible with this system in order to be used as a valid position provider for the navigation module.

B. Map management

The system provides the necessary tools to convert the source maps from OpenStreetMap (i.e., files with the osm extension) into a database. Hence, users can customize their application adding personalized maps. There are two types of scenarios:

- Outdoor maps: OpenStreetMap provides outdoor maps for the whole world. Then, users can download specific regions and add them to the applications as exchangeable maps. For example: a user can download a map of a specific city, a country, a continent, etc. The module is able to manage several maps at the same time. After downloading a specific map, users are allowed to perform local modifications that do not want to upload to the OpenStreetMap's servers. For example, users can modify locally a map to adapt it to a certain kind of social event (e.g., marathons, conferences, expositions, etc.) in order to improve the navigation services. This kind of modifications are not meant to be uploaded to the servers since they are taking place only in a short period of time.

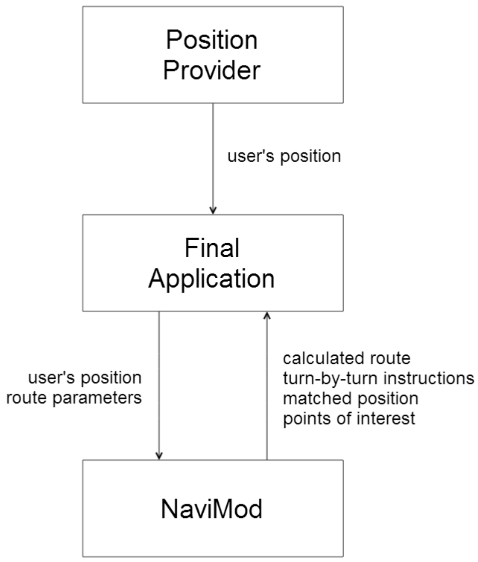

Figure 1. System architecture

- Indoor maps: Due to OpenStreetMap not providing indoor maps, the user has to create his own maps for specific buildings. Unlike the research made by Gotlib, Gnat, and Marciniak [3], we decided not to use a

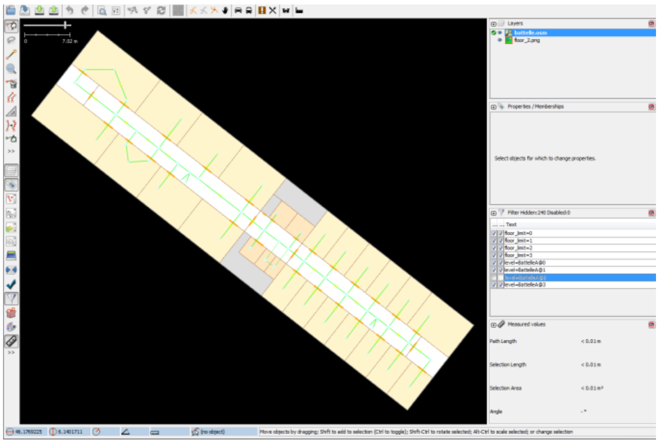

Figure 2. An indoor map design using JOSM

complex format but to adapt the proposal for indoor mapping for OpenStreetMap [4] for future compatibilities. These maps can be shared between users in order to collaborate and improve the navigation service. Once the map is finished, the tool will convert the source map into a database containing the indoor map and the information related to it (i.e., points of interest). Figure 2 shows an example of an indoor map design for the second floor of the University Computing Center at the University of Geneva. The network map was designed using the JOSM[12] tool, which can be used to edit outdoor maps as well. The map contains all the corridors and their connections. Each point of interest (e.g., offices, classrooms, etc.) is represented by a node in the map.

The result of the conversion of the source map is a directed graph that contains all the points and their connections inside the map. This network is stored into a database, called "map network", which will be used to reconstruct the graph and calculate routes.

The reason why the module does not use directly the osm source files is that the database approach offers a better performance accessing the data eliminating all the unnecessary information that is not used by the navigation system.

C. Indoor maps

As previously mentioned, the indoor maps are created using the official OpenStreetMap source format. Therefore, the geographical coordinates related to a node or a point of interest are represented using the same format as the outdoor maps, keeping a strong compatibility between both environments.

OpenStreetMap's format offers a free tagging system that allows the map to contain unlimited data about its elements. A tag consists of a key and a value that are used to describe elements. The community has agreed in a set of standard tags to represent the most common points of interest in a map (e.g., offices, toilets, cafeterias, rooms, elevators, stairs, etc.). Hence, indoor maps can use the same set of tags to represent indoor elements as well. Figure 3 represents a point of interest using standard tags in OpenStreetMap.

```
<node lat='46.1747152' lon='6.1273034'>
    <tag k='ele'         v='430' />
    <tag k='amenity'     v='toilets' />
    <tag k='male'        v='yes' />
    <tag k='wheelchair'  v='yes' />
</node>
```

Figure 3. A point of interest using standard tags in OpenStreetMap

This set of standard tags does not take into account several elements that can be found in a specific indoor environments such as hospitals, museums or airports. However, thanks to the free tagging system, users can define their own tags and then create any kind of point of interest needed. For example, figure 4 represents a printer as a point of interest. This is a common element that can be found often in offices and is not contemplated in the standard set of tags of OpenStreetMap. However, the rules to describe indoor maps are flexible enough to allow users to define and create their own elements for new scenarios.

```
<node lat='46.1766711' lon='6.1397371'>
    <tag k='ele'      v='450' />
    <tag k='machine'  v='printer' />
    <tag k='colour'   v='yes' />
</node>
```

Figure 4. A custom point of interest using OpenStreetMap's format

D. Navigation instructions

The navigation module is able to provide turn-by-turn navigation instructions in order to follow the calculated route. In order to accomplish this task, the module needs the current user's position to perform a technique called map matching.

This technique is used to merge the data from the position provider and the map network to estimate the user's location that best matches the calculated route. The reason that this technique is necessary is that the location acquired from position provider is subject to errors. This feature offers a smoother transition between the different positions acquired by the position provider and avoids unexpected variations in the position.

Once the user's position is matched with the map, the navigation module is able to calculate the next turn that the user needs to perform as well as the distance to it (e.g., turn to the left in 15 meters). The navigation module informs about this event to the main application which will be in charge of display it on the screen.

E. Route algorithm

The module is able to calculate routes between two points that can be separated for a few meters or hundreds of

[12] JOSM, http://josm.openstreetmap.de

kilometers. The route algorithm allows to set the mode of traveling (i.e., by car, by foot, by bike or by wheelchair).

The chosen algorithm to calculate routes used by the module is A*, a generalization of the Dijkstra algorithm, as explained in [5], [6] and [7]. The only difference is that A* uses a heuristic function (also called *h(x)*) in order to speed the algorithm up.

The A* algorithm offers better performance over the Dijkstra algorithm thanks to the heuristic function that allows to "guide" the process in order to find the target node inside the network. The current implemented version of the A* algorithm is able to use two heuristics:

- Distance (Euclidean): for traveling by foot, by bike and by wheelchair to calculate the shortest route
- Time: for traveling by car to calculate the fastest route

Regarding the performance, we encountered a few problems not in the execution of the algorithm itself but loading the information from the database. We faced some latency problems due to managing outdoor maps since the module needs to handle a database of around 500,000 records for a single city.

In table I, we show different measurements related to the performance of the algorithm. Each row contains the average information related to different executions of the calculation of a route. We choose four different scenarios that calculate a route between two points separated by 580 m, 790 m, 2 km and 5 km respectively. The result shows a big time consumption loading the information from the database and a small part of the time running the algorithm to process the loaded nodes and links.

We can also see that as we increase the distance between the start and end point, the ratio between the processed and loaded nodes changes. Considering the last scenario, a total of 13300 nodes where processed over 18300 nodes that where loaded in memory. This means that 72.6 % of the loaded nodes where actually used in the computation of the algorithm. Quantifying the number of nodes in surface area values, 13300 nodes correspond to 10.9 km^2 and 18300 nodes to 15 km^2.

IV. RESULTS

Due to the system consisting only of an Android library, we have created an example application in order to show all the services that the module can provide. Specifically, it consists of an Android application for smartphones and tablets.

As a position provider we used an internal module called GPM (Global Positioning Module) [8], a hybrid positioning framework for mobile device which provides to the final application the user's location, both indoor and outdoor. However, if the final application is meant to be used only in outdoor environments, the position provider could use only the GPS signal.

TABLE I. ALGORITHM PERFORMANCE

Nodes loaded/processed	Total time	Database time		A* time	
640 / 110	165 ms	110 ms	67%	55 ms	33%
900 / 140	260 ms	190 ms	73%	70 ms	27%
4800 / 2300	1240 ms	1070 ms	86%	170 ms	14%
18300 / 13300	5730 ms	5200 ms	91%	530 ms	9%

The example application shows the user's current location and allows calculating routes between two points. The user can select as a start or end point: his current position, a point in a map (touching the screen) or a point of interest from the catalog. In this case we used the Google Maps viewer to show that the navigation module is independent of the map viewer of the final application. It means that the navigation module only provides the services and it is the responsibility of the final application to show these results (in a 2D or 3D map, using augmented reality, voice instructions, etc.).

A. Outdoor

In this case the application works as a standard navigation system (e.g., TomTom) that allows navigating in the city. The example application contains the network map of the city of Geneva. However, if the application is meant to be used in another city, the user just needs to generate the map of the correct region and add it to the application. Figure 5 shows:

- A route by foot from the current user's position to a point of interest in the city. In this case a static route (with the total distance and the estimated time) is displayed to the user, who can accept it and start the navigation or cancel it.
- The user following another route by car. In green the path behind (already done) and in red the path ahead. In this case, the navigation module will monitor the user's position, perform the map matching and provide the correct turn-by-turn directions in real time to guide the user to his destination.

B. Indoor

The example application also contains an indoor map of a building so the user is able to navigate within it. The map network of the building also contains information about the points of interest inside of it (e.g., offices, classrooms, cafeterias, toilets, etc.).

Therefore, a user who enters the building for the first time and needs to reach a specific room can use the application to find it in the points of interest catalog and navigate to it.

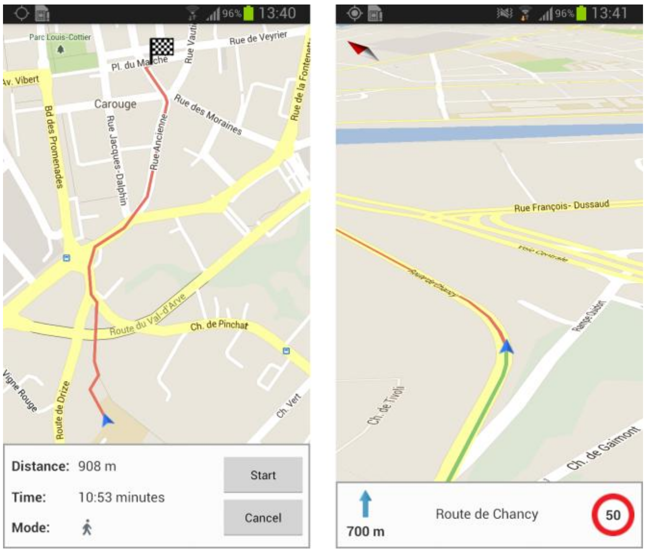

Figure 3. Example of outdoor navigation

The final application is able to show the correct floor plan in each moment using the user's altitude and the map network. Figure 6 shows an indoor path between two rooms and the user navigating by foot thought the same route. In this case the navigation system provides specific turn-by-turn directions for an indoor environment (e.g., no street names or maximum speed indications are shown).

V. CONCLUSIONS AND FUTURE WORK

Thanks to the module implementing all the navigation services, the final application remains small and it is only limited to showing the graphical map, receiving the input parameters from the user and showing the results provided by the navigation system.

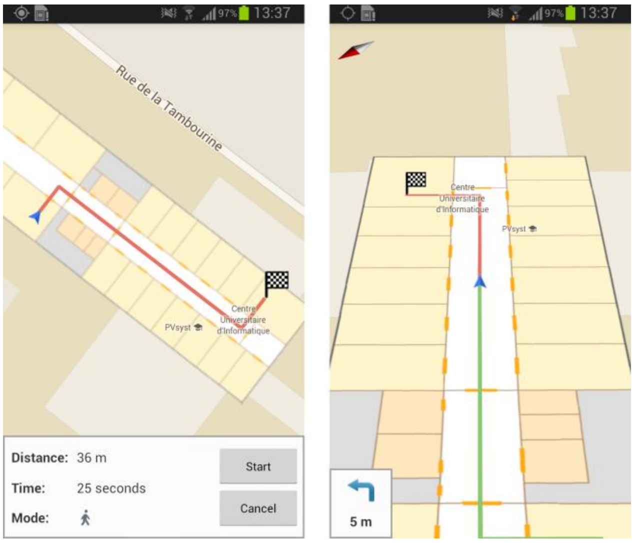

Figure 4. Example of indoor navigation

On the other hand, OpenStreetMap's data does not have enough information to perform geocoding in an outdoor environment. Therefore, is not possible to find the associated geographic coordinates (often expressed as latitude and longitude) from other geographic data, such as street addresses, or postal codes. Due to this problem, for the moment the user is only able to select a starting point or destination by choosing a point of interest from the catalog, selecting a point on the map (touching the screen) or using his current position.

Another problem is that in the current version the user is only able to travel indoor-indoor or outdoor-outdoor. It means that is not possible to travel from an outdoor position to an indoor place or vice versa due to both environments are in two separated maps.

From the performance perspective, the used route algorithm needs to be improved in order to optimize the node processing and reduce the time response. Currently, solutions such as Navfree or OsmAnd implement algorithms that compute the route in half of the time. This is an important point to take into account for future improvements of the module.

Additionally we are looking into creating maps connections to calculate routes between indoor and outdoor environments. Furthermore, we planned to work on the taxonomy of indoor maps to offer a better indoor navigation service.

ACKNOWLEDGMENT

This work is supported by the AAL Virgilius Project (aal-2011-4-046).

REFERENCES

[1] Yali Jiang, Yuan Fang, Chunlong Yao, and Zhisen Wang, "A design of indoor & outdoor navigation system", proceedings of ICCTA 2011.

[2] Hui Li, and Xiangyang Gong, "An approach to integrate outdoor and indoor maps for books navigation on the intelligent mobile device" IEEE 3rd International Conference on Communication Software and Networks (ICCSN), 2011.

[3] Dariusz Gotlib, Milosz Gnat, and Jacek Marciniak, "The Research on Cartographical Indoor Presentation and Indoor Route Modeling for Navigation Applications", International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012.

[4] Indoor mapping proposal for OpenStreetMap, as on October 2013, http://wiki.openstreetmap.org/wiki/Indoor_Mapping

[5] Ioannis Kaparias, and Michael G. H. Bell, "A Reliability-Based Dynamic Re-Routing Algorithm for In-Vehicle Navigation", Annual Conference on Intelligent Transportation Systems 2010.

[6] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Transactions of systems science and cybernetics, 1968.

[7] T.M. Rao, Sandeep Mitra, and James Zollweg, "Snow-Plow Route Planning using AI Search" 2011.

[8] Anja Bekkelien, and Michel Deriaz, "Hybrid Positioning Framework for Mobile Devices", 2nd International Conference on Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), 2012.